// Pentru a rula acest cod, asigurați-vă că aveți un mediu React funcțional.  
// Tailwind CSS este încărcat printr-un link CDN, adăugați acest link în <head>-ul fișierului HTML:  
// <script src="https://cdn.tailwindcss.com"></script>  
  
import React, { useState } from 'react';  
  
// --- Iconițe SVG pentru un design curat și independent ---  
const PlusIcon = () => (<svg xmlns="http://www.w3.org/2000/svg" className="h-5 w-5 mr-2" viewBox="0 0 20 20" fill="currentColor"><path fillRule="evenodd" d="M10 3a1 1 0 011 1v5h5a1 1 0 110 2h-5v5a1 1 0 11-2 0v-5H4a1 1 0 110-2h5V4a1 1 0 011-1z" clipRule="evenodd" /></svg>);  
const RocketIcon = () => (<svg xmlns="http://www.w3.org/2000/svg" className="h-5 w-5 mr-2" viewBox="0 0 20 20" fill="currentColor"><path d="M10.894 2.553a1 1 0 00-1.788 0l-7 14a1 1 0 001.169 1.409l5-1.429A1 1 0 009 15.571V11a1 1 0 112 0v4.571a1 1 0 00.725.962l5 1.428a1 1 0 001.17-1.408l-7-14z" /></svg>);  
const DownloadIcon = () => (<svg xmlns="http://www.w3.org/2000/svg" className="h-5 w-5 mr-2" fill="none" viewBox="0 0 24 24" stroke="currentColor" strokeWidth={2}><path strokeLinecap="round" strokeLinejoin="round" d="M4 16v1a3 3 0 003 3h10a3 3 0 003-3v-1m-4-4l-4 4m0 0l-4-4m4 4V4" /></svg>);  
const SpinnerIcon = () => (<svg className="animate-spin -ml-1 mr-3 h-5 w-5 text-white" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24"><circle className="opacity-25" cx="12" cy="12" r="10" stroke="currentColor" strokeWidth="4"></circle><path className="opacity-75" fill="currentColor" d="M4 12a8 8 0 018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824 3 7.938l3-2.647z"></path></svg>);  
  
  
// --- Componenta pentru Antet ---  
const AppHeader = ({ workshopTitle, onGenerate, isLoading, downloadUrl }) => {  
 return (  
 <header className="bg-white shadow-sm p-3 flex items-center justify-between z-10 sticky top-0">  
 <div className="flex items-center">  
 <div className="bg-indigo-600 text-white font-bold rounded-md p-2 mr-4 text-sm">SF</div>  
 <h1 className="text-lg font-semibold text-gray-800 truncate">{workshopTitle}</h1>  
 </div>  
 <div className="flex items-center space-x-3">  
 <span className="text-sm text-gray-500 italic">Salvat ✓</span>  
 {downloadUrl ? (  
 <a href={downloadUrl} download className="flex items-center text-sm bg-green-500 text-white font-semibold py-2 px-4 rounded-lg hover:bg-green-600 transition duration-300 shadow-md">  
 <DownloadIcon />  
 Descarcă Pachetul  
 </a>  
 ) : (  
 <button   
 onClick={onGenerate}   
 disabled={isLoading}  
 className="flex items-center text-sm bg-indigo-600 text-white font-semibold py-2 px-4 rounded-lg hover:bg-indigo-700 transition duration-300 shadow-md disabled:bg-indigo-400 disabled:cursor-not-allowed"  
 >  
 {isLoading ? <SpinnerIcon /> : <RocketIcon />}  
 {isLoading ? 'Se generează...' : 'Generează & Împachetează'}  
 </button>  
 )}  
 </div>  
 </header>  
 );  
};  
  
// --- Componentele de UI (ConfigurationPanel, AssemblyArea, LivePreview) rămân neschimbate ---  
const ConfigurationPanel = ({ workshop, setWorkshop }) => {  
 const handleInputChange = (e) => {  
 const { id, value } = e.target;  
 setWorkshop(prev => ({ ...prev, [id]: value }));  
 }  
 return (  
 <div className="bg-white p-6 rounded-xl shadow-lg h-full">  
 <h2 className="text-xl font-bold text-gray-800 mb-6 border-b pb-3">1. Planul Atelierului</h2>  
 <div className="space-y-6">  
 <div>  
 <label htmlFor="title" className="block text-sm font-medium text-gray-700 mb-1">Titlul Atelierului</label>  
 <input type="text" id="title" value={workshop.title} onChange={handleInputChange} className="w-full px-3 py-2 border border-gray-300 rounded-lg focus:ring-indigo-500 focus:border-indigo-500 transition" placeholder="ex: Micii Exploratori Polari" />  
 </div>  
 <div>  
 <label htmlFor="theme" className="block text-sm font-medium text-gray-700 mb-1">Tema Principală</label>  
 <textarea id="theme" rows="3" value={workshop.theme} onChange={handleInputChange} className="w-full px-3 py-2 border border-gray-300 rounded-lg focus:ring-indigo-500 focus:border-indigo-500 transition" placeholder="ex: Animale polare, iarnă, gheață, adaptări la frig"></textarea>  
 </div>  
 <div>  
 <label className="block text-sm font-medium text-gray-700 mb-2">Vârsta Țintă</label>  
 <div className="flex space-x-2">  
 <button className="text-sm flex-1 bg-indigo-100 text-indigo-700 font-semibold py-2 px-3 rounded-lg hover:bg-indigo-200 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500">1-3 ani</button>  
 <button className="text-sm flex-1 bg-white text-gray-700 font-semibold py-2 px-3 rounded-lg border hover:bg-gray-50">3-5 ani</button>  
 <button className="text-sm flex-1 bg-white text-gray-700 font-semibold py-2 px-3 rounded-lg border hover:bg-gray-50">5-7 ani</button>  
 </div>  
 </div>  
 </div>  
 </div>  
 );  
};  
const AssemblyArea = ({ workshop, setWorkshop }) => {  
 const addComponent = (type, subTheme) => {  
 const newComponent = { id: Date.now(), type: type, subTheme: subTheme };  
 setWorkshop(prev => ({ ...prev, components: [...prev.components, newComponent] }));  
 }  
 return (  
 <div className="bg-white p-6 rounded-xl shadow-lg h-full flex flex-col">  
 <h2 className="text-xl font-bold text-gray-800 mb-6 border-b pb-3">2. Componentele Atelierului</h2>  
 <div className="flex-grow bg-gray-50 rounded-lg p-4 border-dashed border-2 border-gray-300 text-center overflow-y-auto">  
 {workshop.components.length === 0 ? (<p className="text-gray-500 mt-4">Atelierul tău este gol.</p>) : (  
 <ul className="text-left space-y-3">  
 {workshop.components.map(comp => (  
 <li key={comp.id} className="bg-white p-3 rounded-lg shadow-sm flex justify-between items-center">  
 <span className="font-medium text-gray-700">{comp.type}: <span className="text-gray-500">{comp.subTheme}</span></span>  
 </li>  
 ))}  
 </ul>  
 )}  
 </div>  
 <div className="mt-4 flex space-x-2">  
 <button onClick={() => addComponent('Pagină de Colorat', 'Robot Prietenos')} className="flex-1 flex items-center justify-center bg-indigo-500 text-white font-semibold py-3 px-4 rounded-lg hover:bg-indigo-600 transition duration-300"><PlusIcon />Pagină de Colorat</button>  
 <button onClick={() => addComponent('Fișă de Trasare', 'Litere R-O-B-O-T')} className="flex-1 flex items-center justify-center bg-purple-500 text-white font-semibold py-3 px-4 rounded-lg hover:bg-purple-600 transition duration-300"><PlusIcon />Fișă de Trasare</button>  
 </div>  
 </div>  
 );  
};  
const LivePreview = ({ workshop }) => {  
 return (  
 <div className="bg-white p-6 rounded-xl shadow-lg h-full">  
 <h2 className="text-xl font-bold text-gray-800 mb-6 border-b pb-3">3. Previzualizare Ghid</h2>  
 <div className="bg-gray-100 p-6 rounded-lg h-5/6 overflow-y-auto font-serif">  
 <h1 className="text-3xl font-bold text-center mb-4">{workshop.title}</h1>  
 <p className="text-gray-700 mb-6 italic">O poveste magică de introducere despre <strong>{workshop.theme || 'o temă minunată'}</strong>...</p>  
 {workshop.components.map((comp, index) => (  
 <div key={comp.id}>  
 <h3 className="text-xl font-semibold mt-6 mb-2 border-t pt-4">Activitatea {index + 1}: {comp.type}</h3>  
 <div className="w-full h-40 bg-white rounded-lg border-2 border-gray-300 flex items-center justify-center"><p className="text-gray-400">[Previzualizare: {comp.subTheme}]</p></div>  
 </div>  
 ))}  
 </div>  
 </div>  
 );  
};  
  
// --- Componenta Principală a Aplicației ---  
export default function App() {  
 const [workshop, setWorkshop] = useState({  
 title: 'Atelier Nou Fără Titlu',  
 theme: 'Roboți și Tehnologie',  
 target\_age: '3-5 ani',  
 components: [],  
 });  
   
 const [isLoading, setIsLoading] = useState(false);  
 const [error, setError] = useState(null);  
 const [downloadUrl, setDownloadUrl] = useState(null);  
  
 // --- Funcția REALĂ care face apelul către backend ---  
 const handleGenerateWorkshop = async () => {  
 setIsLoading(true);  
 setError(null);  
 setDownloadUrl(null);  
   
 // Acesta este URL-ul public al API Gateway-ului pe care l-am proiectat.  
 // El va fi înlocuit cu URL-ul real odată ce Gateway-ul este implementat.  
 const apiGatewayUrl = 'https://sparky-gateway-xxxx.ew.gateway.dev/v1/generate-workshop';  
  
 try {  
 console.log("Se trimit următoarele date către backend:", workshop);  
   
 const response = await fetch(apiGatewayUrl, {  
 method: 'POST',  
 headers: {  
 'Content-Type': 'application/json',  
 // Aici s-ar adăuga token-ul de autentificare  
 // 'Authorization': `Bearer ${userAuthToken}`  
 },  
 body: JSON.stringify(workshop)  
 });  
  
 if (!response.ok) {  
 // Dacă răspunsul de la server este o eroare (ex: 404, 500)  
 const errorData = await response.json();  
 throw new Error(errorData.message || 'A apărut o eroare de la server.');  
 }  
  
 const result = await response.json();  
   
 // Ne așteptăm ca backend-ul să returneze un obiect JSON care conține URL-ul de descărcare  
 // ex: { downloadUrl: 'https://storage.googleapis.com/...' }  
 console.log("Răspuns de succes de la backend:", result);  
 setDownloadUrl(result.downloadUrl);  
  
 } catch (err) {  
 console.error("Eroare la apelul API:", err);  
 setError(err.message || 'Nu s-a putut contacta serverul. Vă rugăm verificați conexiunea la internet.');  
 } finally {  
 setIsLoading(false);  
 }  
 };  
  
 return (  
 <div className="bg-gray-50 font-sans h-screen flex flex-col">  
 <AppHeader   
 workshopTitle={workshop.title}   
 onGenerate={handleGenerateWorkshop}   
 isLoading={isLoading}  
 downloadUrl={downloadUrl}  
 />  
  
 {error && (  
 <div className="bg-red-100 border-l-4 border-red-500 text-red-700 p-4 m-4" role="alert">  
 <p className="font-bold">Eroare</p>  
 <p>{error}</p>  
 </div>  
 )}  
  
 <main className="flex-grow p-4 md:p-8 grid grid-cols-1 lg:grid-cols-3 gap-8">  
 <div className="lg:col-span-1">  
 <ConfigurationPanel workshop={workshop} setWorkshop={setWorkshop} />  
 </div>  
 <div className="lg:col-span-1">  
 <AssemblyArea workshop={workshop} setWorkshop={setWorkshop} />  
 </div>  
 <div className="lg:col-span-1">  
 <LivePreview workshop={workshop} />  
 </div>  
 </main>  
 </div>  
 );  
}